MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE

NASA Communications (Nascom) Small Conversion Device (SCD) Installation and Troubleshooting Guide

November 1996

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland

Contents

Section 1. Introduction

Section 2. User Logins

Section 3. Ethernet Board Configuration

Section 4. AMI BIOS Configuration

Section 5. Network Configuration

Section 6. System Diagnostics

Section 7. SCD NIB Configuration

Section 8. SCD Software Updates

Section 9. System Software Updates

Section 10. SCD Test Tools

Section 11. Disk Drive Rescue

Appendix A - Parts List

Figures

Section 1. Introduction

1.1 Purpose

This NASA Communications (Nascom) Small Conversion Device (SCD) Installation and Troubleshooting Guide provides instructions for installation of the SCD.

1.2 Small Conversion Device Description

1.2.1 SCD Hardware Description

The SCD is a PC-based system which will accept 4800-bit block serial input and convert it to IP-based output, and conversely will accept IP-based input and transmit 4800-bit serial blocks to the end user. It is intended to operate similar to a "black box" with minimal operator intervention. Its configuration will be provided and controlled by the Nascom Network Management System (NMS).

The SCD implementation is centered around the Nascom Interface Board (NIB), a serial line card, which is also used in operational systems including the Nascom Tracking Data System (TDS). The NIB has an EIA-449 connector to receive and transmit 4800-bit blocks. The SCD also has an Ethernet connection to support UDP/IP communications.

1.2.2 SCD Software Description

The SCD software consists of a control (SCNTR) process which starts other processes based on the configuration file and an operator interface (SOI) process which is started when logging into one of the captive SCD accounts. The following processes will be started by SCNTR based on the SCD configuration file:

- Serial output (SSERO) - receives de-encapsulated 4800-bit block messages from the UDP input process and writes them to the NIB card (a SSERO process is started for each NIB card that is configured as present and enabled for writing)
- Serial input (SSERI) - reads 4800-bit block messages from the NIB card and sends them to either the Fixed or Table UDP output process (based on the configuration file) to be encapsulated and transmitted out the Ethernet card (a SSERI process is started for each NIB card that is configured as present and enabled for reading)
- UDP input (SUDPI) - reads the UDP encapsulated 4800-bit block messages and sends them to the specified serial output process based on joined group information specified in the SCD configuration file
- Fixed UDP output (SUDPO) - receives 4800-bit block messages from a serial input process and encapsulates them in UDP messages and transmits them out the Ethernet card to the specified IP address (this process does not look at the destination code in the Nascom header)

- Table UDP output (SUDPO) - receives 4800-bit block messages from a serial input process and encapsulates them in UDP messages and transmits them out the Ethernet card to the specified IP address (the IP address is determined by looking up the destination code, specified in the Nascom header, in the MSS routing table)
- Build configuration table (SCFG) - reads the SCD configuration and MSS routing table files and builds the internal shared memory tables and then notifies the SCD control process of any table changes
- SNMP Agent (SNMP-A) - Not Yet Implemented
- SNMP Statistics (SSTATS) - Not Yet Implemented

Figure 1-1 shows the interaction between the various SCD processes.

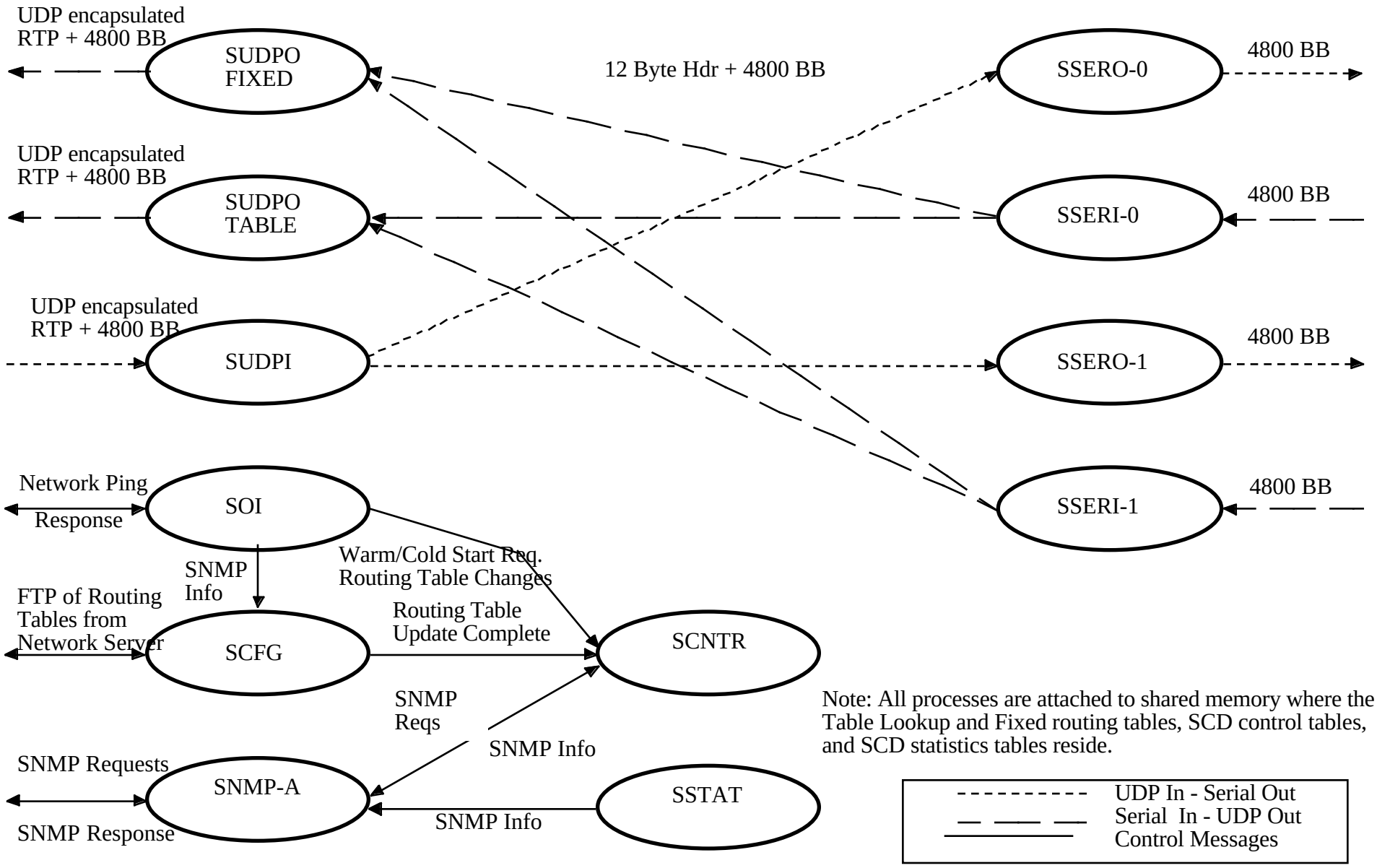


Figure 1.1 SCD Processes

1.3 Organization

Following the Introduction (Section 1), this document presents procedures and reference material on the specified topics in the following order:

a. User Logins (Section 2)

b. Ethernet Board Configuration (Section 3)

c. AMI BIOS Configuration (Section 4)

d. Network Configuration (Section 5)

e. System Diagnostics (Section 6)

f. SCD NIB Configuration (Section 7)

g. SCD Software Updates (Section 8)

h. System Software Updates (Section 9)

i. SCD Test Tools (Section 10)

1.4 Reference Documents

a. *NASA Communications (Nascom) Small Conversion Device (SCD) Requirements Document*, June 1996.

b. *NASA Communications (Nascom) Internet Protocol (IP) Transition Data Format Document*, June 1996.

c. *NASA Communications (Nascom) Small Conversion Device (SCD) Operator's Guide*, June 1996.

d. *Nascom IP Transition Project Transition Plan*, June 1996.

Section 2. User Logins

2.1 Introduction

To login to the SCD, the operator can either use a console attached to the SCD or telnet to the SCD using a package that supports VT100 emulation. The SCD provides two types of user logins: installation and operational. The installation logins are for setting up and testing the SCD when it is installed at the site. The operational logins are for managing the SCD during actual operations.

2.1.1SCD Installation Logins

Three installation login accounts are provided: **root**, **scdinstl**, and **scdtest**. The root account is required to run netconfig (see Section 5.1 for details) and load new system software (see Section 9.1 for details). The scdinstl and scdtest accounts are used to run various test programs to generate test data or monitor data flows (see Sections 5.1 and 10.1 for details).

2.1.2SCD Operational Logins

Three operational login accounts are provided: **scdops**, **scdadmin**, and **scdcm**. The scdops and scdadmin accounts have captive login shells (i.e. no access to operating system commands) that will start the SCD Operator Interface process (SOI). In the case of the scdadmin account, an administrative privilege flag is set to allow the operator to modify the SCD configuration (see Section 7.1 for details on NIB configuration). The scdcm account has a captive login shell (i.e. no access to operating system commands) that will provide a CM menu to load the SCD software, load the SCD configuration file, or create a SCD configuration backup (see Section 8.1 for details).

Section 3. Ethernet Board Configuration

3.1 Ethernet Board Configuration

The Ethernet board is supposed to automatically configure itself, but there have been problems with 3c90x boards using the AUI port not detecting the AUI connection. A DOS bootable diskette has been created which contains utilities for configuring and validating the boards.

In order to be able to boot from the A drive, change the AMI BIOS Advanced menu item BootUp Sequence to "A:,C:,CD-ROM". See Section 4 for details.

Insert the DOS diskette containing the Ethernet diagnostic utilities into drive A and reboot the machine. A menu of utilities will appear.

To configure a board connected to the Ethernet via its AUI port, select the utility which will auto-configure the board to AUI.

To configure a board connected to the Ethernet via either its thinnet or 10BaseT/UTP port, select the utility which will auto configure the board using the default/standard configuration.

To validate and test the board, select the 3Com Ethernet utility. Use the View option to check the current settings. (Note: the cursor moves very slowly in this utility). Items of particular importance to the SCD operations are shown in **boldface**. Items which may vary from SCD to SCD are shown as "...".

View

Adapter Information...

Device/Slot number:	…	
Bus number:	…	
I/O port address:	…	
Interrupt request level:	**10** or **11**	(not 3 or 4)
Media type:	**Auto Select (N-Way)**	(thinnet or UTP)
	AUI	(AUI)
Boot PROM size:	Disabled	
Network speed:	**10M Bit/s**	
Receive FIFO size:	**3k**	(not zero)
Transmit FIFO size:	**5k**	(not zero)

Interface product revision:	…
Date of Manufacture:	…
Division Code: …	
Product Code:	…
Network driver optimization:	**Maximized Network Performance**
Full Duplex:	**Disabled**

To change the settings, use the Install menu option. A combination of tab keys, arrow keys, and the return key are required to select and change entries.

To test the board, use the Test menu option. All tests should complete successfully.

After the setup is complete, reboot and reset AMI BIOS BootUp Sequence to "C:,A:,CD-ROM" so that the SCD will successfully boot with a configuration diskette in drive A.

Section 4. AMI BIOS Configuration

4.1 AMI BIOS Configuration For Rack Mountable SCDs

When the system is powered on, press the Del key to enter the AMI BIOS setup.

AMI BIOS Setup Info for TYAN S-1468 Motherboard:

Items of particular importance to the SCD operations are shown in **boldface**.

To auto detect the IDE drives that are installed, go to the Utility window and activate the Detect IDE icon.

Standard:

Primary Master	[The system will usually detect appropriate settings for the drive as formatted.]	
Type	USER	
LBA/Large Mode	ON	
Block Mode	ON	
32 Bit Mode	OFF	
PIO Mode	3	
Primary Slave		
Type	NOT INSTALLED	CDROM (INSTALLED)
LBA/Large Mode	N/A	ON
Block Mode	N/A	ON
32 Bit Mode	N/A	OFF
PIO Mode	N/A	AUTO
Secondary Master and Secondary Slave		
Type	NOT INSTALLED	
Floppy A	**1 44 MB 3 1/2**	
Floppy B	NOT INSTALLED	

Advanced:

Quick Boot	Enabled
BootUp Sequence	**C:, A:, CD-ROM**
BootUp NumLock	ON
Floppy Drive Swap	DISABLED
Mouse Support	DISABLED
Primary Display	VGA/EGA
Password Check	SETUP
OS/2 Compatible Mode	DISABLED
Internal Cache	**WRITEBACK**
External Cache	**ENABLED**
System BIOS Cacheable	ENABLED
C000, 16K Shadow	ENABLED
C400, 16k Shadow	ENABLED
All other addresses	DISABLED

Chipset:

Memory Hole	DISABLED	
DRAM Speed	60 ns	[may need 70ns if lockups occur]
IRQ 12/M Mouse Function	**DISABLED**	
8 Bit I/O Recovery	**1**	
16 Bit I/O Recovery	**1**	

Power management:

DISABLED for now. Will be testing monitor/HDD functions for impact.

PCl/PnP Setup:

Plug and Play Aware OS	NO
PCI Burst Mode	ENABLED
PCI Concurrency	ENABLED
PCI Streaming	ENABLED
PCI Latency	64
PCI Palette Snoop	DISABLED

PCI IDE Busmaster	DISABLED
Off board PCI IDE Card	AUTO
	SCD
IRQ 3	**ISA/EISA**
IRQ 4	**ISA/EISA**
IRQ 5	PCI/PnP
IRQ 7	PCI/PnP
IRQ 9	PCI/PnP
IRQ l0	PCI/PnP
IRQ 11	PCI/PnP
IRQ 12	PCI/PnP
IRQ 14	PCI/PnP
IRQ 15	**ISA/EISA**
Reserved Memory Size	DISABLED

Peripheral Setup:

Onboard FDC	AUTO
Onboard Serial Port 1	AUTO
Onboard Serial Port 2	AUTO
Onboard Parallel Port	AUTO
Parallel Port Mode	NORMAL
Onboard PCI IDE	PRIMARY

4.3 TYAN S-1468 Motherboard Jumper Settings

Jumper	Position	Location	Function
J24	**3-4 120MHz** 1-2 133 Mhz	**By RTC**	**CPU P55C**
J41	open	By RTC	CMOS erase
J13	open	middle	COAST cache
J23		middle	
J35	OFF	CPU Jumper bank	3.48V
J36	OFF	CPU Jumper bank	3.48V
J37	OFF	CPU Jumper bank	3.48V
J38	OFF	CPU Jumper bank	3.48V
J39	OFF	CPU Jumper bank	3.48V
J40	OFF	CPU Jumper bank	3.48V

J54	ON	CPU Jumper bank	Static
J53	OFF	CPU Jumper bank	Static
J55	ON	CPU Jumper bank	Static
J56	ON	CPU Jumper bank	Static
J26	ON	CPU Jumper bank	Static
J27	ON	CPU Jumper bank	Static
J28	OFF	CPU Jumper bank	Cache/static
J29	OFF	CPU Jumper bank	Cache/static
J30	OFF	CPU Jumper bank	Cache/static
J31	OFF	CPU Jumper bank	Cache/static
J32	**ON**	**CPU Jumper bank**	**CPU P55C**
J33	**OFF**	**CPU Jumper bank**	**CPU P55C**
J1	ON	floppy connector	5 V EDO Mem
J2	ON	floppy connector	5 V EDO Mem
J9	ON	memory bank 1	5 V EDO Mem
J11	ON	memory bank 1 and 0	5 V EDO Mem
J4	OFF	bank 0	5 V EDO Mem
J5	OFF	bank 0	5 V EDO Mem
J6	OFF	bank 0	5 V EDO Mem
J45	connector	CPU corner	speaker
J46	connector	CPU corner	keylock
J47	1-2	CPU corner	turbo
J48	connector	CPU corner	HDD LED
J49	connector	CPU corner	reset
J50		CPU corner	turbo LED
J42	1-2	BIOS / ISA	EPROM type
J43	1-2	BIOS / ISA	EPROM type
J44	1-2	BIOS / ISA	EPROM type
J25	**1-2 SMC 665IR** 2-3 SMC 669IR	ISA slot 3	IO chip
J14		ISA slot 2	
J19		ISA slot 2	
J20		ISA slot 2	
J15		ISA slot 2	
J16		ISA slot 2	
J21		ISA slot 2	
J22	**1-2**	**ISA slot 2**	**10MHz AT clock**
J17		ISA slot 2	
J18		ISA slot 2	

J51	1-2	PCI slot 2	IR IO
J52	1-2	PCI slot 2	IR IO
J12		edge by PCI slot 2	

4.3 AMI BIOS Configuration For Desktop/Tower SCDs

When the system is powered on, press the F1 key to enter the AMI BIOS setup.

AMI BIOS Setup information for WIN Pentium Model 586-iXXev-pci Motherboard:

Items of particular importance to the SCD operations are shown in **boldface**.

Main:

Primary Master — [The system will usually detect appropriate settings for the drive as formatted.]

Primary Slave

Type — NOT INSTALLED

Secondary Master and Secondary Slave

Type — NOT INSTALLED

Floppy Options:

Floppy A — **1.44 MB 3 1/2”**

Floppy B — NOT INSTALLED

Boot Options:

First: — **Hard Disk**

Second: — **Floppy**

Third: — Disabled

Fourth: — Disabled

System Cache:

Boot Speed: — **Turbo**

Num Lock: — Off

Setup Prompt: — Enabled

Hard Disk Pre-Delay:	Disabled
Typematic Rate Prog:	Default

Advanced:

Peripheral Configuration:

Configuration Mode:	**Manual**
Primary PCI IDE I/F:	Enabled
Secondary PCI IDE I/F:	**Enabled**
Floppy I/F:	**Enabled**
Serial Port 1 Address:	**Disabled**
Serial Port 2 Address:	**Disabled**
Parallel Port Address:	**Disabled**
Parallel Port Mode:	Compatible

Audio Configuration:

Configuration Mode:	**Disabled**

Advanced Chipset Configuration:

Base Memory Size:	640K
ISA LFB Size:	Disabled
Video Palette Snoop:	Disabled
Latency Timer (PCI Clock):	64
PCI Burst:	Enabled

Power Management Configuration:

Advanced Power Mgmt:	Disabled

Plug and Play Configuration:

Configuration Mode:	**Use Setup Utility**
ISA Shared Memory Size:	Disabled
IRQ 3:	**Available**
IRQ 4:	**Available**

IRQ 5: **Available**
IRQ 7: **Available**
IRQ 9: **Available**
IRQ 10: **Available**
IRQ 11: **Available**
IRQ 12: **Available**

Security:

User Password is Disabled

Administrative Password is Disabled

4.4 WIN Pentium Model 586-iXXev-pci Motherboard Switch Settings

There are eight (8) dip switches on the motherboard that must be set.

Switch Function	ON	OFF	NUM
CPU Clk Select		OFF	8
CPU Clk Select		OFF	7
Bus Frequency	**ON**		6
Setup Disable		OFF	5
Clear CMOS		OFF	4
Clear Password		OFF	3
VRE Enable		OFF	2
ISA Clk Speed	**ON**		1

Section 5. Network Configuration

5.1 Netconfig

Once the CMOS has been configured and the system has been booted, netconfig must be run to configure the TCP/IP information. Log in as **root** and enter the required information at the prompt. The following is an example of running netconfig (user input is in **bold**):

>**netconfig**

NETWORK CONFIGURATION

Now we will attempt to configure your mail and TCP/IP. This process probably

won't work on all possible network configurations, but should give you a good

start. You will be able to reconfigure your system at any time by typing:

netconfig

First, we'll need the name you'd like to give your host. Only

the base hostname is needed right now. (not the domain)

Enter hostname: **scdb**

Now, we need the domain name. Do not supply a leading '.'

Enter domain name for scdb: **ops.nascom.nasa.gov**

If you only plan to use TCP/IP through loopback, then your IP address will

be 127.0.0.1 and we can skip a lot of the following questions.

Do you plan to ONLY use loopback ([y]es, [n]o)? **n**

Enter your IP address for the local machine. Example: 111.112.113.114

Enter IP address for scdb (aaa.bbb.ccc.ddd): **192.168.201.114**

Enter your gateway/router address, such as 111.112.113.1

or you can probably get away with entering your own IP address here.

Enter gateway address (aaa.bbb.ccc.ddd): **192.168.201.161**

Enter your netmask. This will generally look something like this: 255.255.255.0

Enter netmask (aaa.bbb.ccc.ddd): **255.255.255.0**

Setting up TCP/IP...

Creating /etc/HOSTNAME...

Creating /etc/rc.d/rc.inet1...

Creating /etc/networks...

Creating /etc/hosts...

Will you be accessing a nameserver ([y]es, [n]o)? **n**

Your networking software has now been configured.

You must reboot the system for the changes to take effect.

> **init 6** (this will shut the system down nicely and then reboot)

5.2 Netconfig Verification

After the system has booted, you can verify the information by logging in as **scdtest** or **scdinstl** and using the route and ifconfig commands.

scdb:> **/sbin/route**

Kernel routing table

Destination	Gateway	Genmask	Flags	MSS	Window		Use	Iface
localnet eth0	*	255.255.255.0		U	1500	0		70
loopback	*	255.0.0.0	U	3584	0		9	lo

```
default         192.168.201.161     *               UG      1500    0               44      eth0
```

```
scdb:> /sbin/ifconfig
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
        UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
        RX packets:216 errors:0 dropped:0 overruns:0
        TX packets:216 errors:0 dropped:0 overruns:0

eth0    Link encap:10Mbps Ethernet  HWaddr 00:A0:24:80:2C:C9
        inet addr:192.168.201.111  Bcast:192.168.201.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:33964840 errors:3259 dropped:3259 overruns:3565
        TX packets:913255 errors:0 dropped:0 overruns:0
        Interrupt:11 Base address:0xef80
```

```
scdb:> /sbin/ping 150.100.100.100
```

If the wrong IP address is indicated, run the netconfig procedure again and re-boot. The results of the procedure are applied to a system startup file which can be reviewed for correctness ‘more /etc/rc.d./rc.inet1’.

If ifconfig lists only the local loopback interface, then the Ethernet card failed to be located or initialized by the Linux kernel. Check the system startup messages using ‘**dmesg|more**’ (Section 6.1).

Possibly, the Ethernet card selected an IRQ used by another device such as a NIB card. If so, check the AMI BIOS settings again (Section 4.1), and be sure to set IRQ3 and 4 to ‘ISA/EISA’ to prevent the Ethernet card from allocating these IRQs.

Possibly the Ethernet card is not installed correctly, or it is not a card type and model recognized by the Linux kernel as configured and delivered.

If the ifconfig listing indicates that the card is configured, but the ping cannot reach the nearest router, it is possible that the Ethernet card failed to Autoselect the correct interface. Cards with an AUI adapter and an RJ45 UTP connection may fail to select the AUI properly. This can be checked

by reviewing the system startup messages using **dmesg** (Section 6.1). If the system failed to select the correct interface, then the Ethernet card must be manually configured for the AUI connection using the stand-alone DOS Bootable diskette provided and the appropriate Ethernet utility (Section 3.1).

Section 6. System Diagnostics

6.1 System Boot Information

After the system has booted, you can verify the system boot information by logging in as **scdtest** or **scdinstl** and using the dmesg command. The information in **bold** shows system memory, OS version, NIB cards and addresses, floppy drive, and Ethernet card.

scdb:> **dmesg | more**

Console: 16 point font, 400 scans

Console: color VGA+ 80x25, 1 virtual console (max 63)

pcibios_init : BIOS32 Service Directory structure at 0x000fdb50

pcibios_init : BIOS32 Service Directory entry at 0xfdb60

pcibios_init : PCI BIOS revision 2.10 entry at 0xfdb81

Probing PCI hardware.

Calibrating delay loop.. ok - 47.92 BogoMIPS

Memory: 24804k/**32768k available** (700k kernel code, 384k reserved, 6880k data)

Swansea University Computer Society NET3.034 for Linux 1.3.77

NET3: Unix domain sockets 0.12 for Linux NET3.033.

Swansea University Computer Society TCP/IP for NET3.034

IP Protocols: IGMP, ICMP, UDP, TCP

Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.

Checking 'hlt' instruction... Ok.

Linux version 2.0.12 (root@scda) (gcc version 2.7.2) **#41 Tue Sep 3 14:26:15 EDT** 1996

Serial driver version 4.13 with no serial options enabled

tty00 at 0x03f8 (irq = 4) is a 16550A

tty01 at 0x02f8 (irq = 3) is a 16550A

lp1 at 0x0378, (polling)

NASCOM Interface Board(NIB) Driver Release '96.2' for Linux

NIB spl_level=6 spl=109890

NIB0 base=220 irq=3 major=42

loop: registered device at major 7

ide: 430FX (Triton) on PCI bus 0 function 57

ide0: BM-DMA at 0xffa0-0xffa7

ide0 timing: (0xa107) sample_CLKs=3, recovery_CLKs=3

master: fastDMA=on PreFetch=on IORDY=on fastPIO=on

slave : fastDMA=off PreFetch=off IORDY=off fastPIO=off

ide1: BM-DMA at 0xffa8-0xffaf

ide1 timing: (0x8000) sample_CLKs=5, recovery_CLKs=4

master: fastDMA=off PreFetch=off IORDY=off fastPIO=off

slave : fastDMA=off PreFetch=off IORDY=off fastPIO=off

hda: ST51080A, 1033MB w/128kB Cache, LBA, CHS=413/64/63

ide0 at 0x1f0-0x1f7,0x3f6 on irq 14

hdc: MATSHITA CR-581, ATAPI CDROM drive

ide1 at 0x170-0x177,0x376 on irq 15

Floppy drive(s): fd0 is 1.44M

Started kswapd v 1.4.2.2

FDC 0 is a post-1991 82077

scsi : 0 hosts.

scsi : detected total.

lance.c: PCI bios is present, checking for devices...

Overriding PCI latency timer (CFLT) setting of 64, new value is 255.

eth0: 3Com 3c590 Vortex 10Mbps at 0xef80, 00:a0:24:80:2c:c9, IRQ 11

8K byte-wide RAM 1:1 Rx:Tx split, autoselect/10baseT interface.

3c59x.c:v0.13 2/13/96 becker@cesdis.gsfc.nasa.gov

Partition check:

hda: hda1 hda2

VFS: Mounted root (ext2 filesystem) readonly.

Adding Swap: 66524k swap-space

VFS: Disk change detected on device 02:28

hdc: media changed

VFS: Disk change detected on device 16:00

Max size:331555 Log zone size:2048

First datazone:68 Root inode number 139264

ISO9660 Extensions: RRIP_1991A

Section 7. SCD NIB Configuration

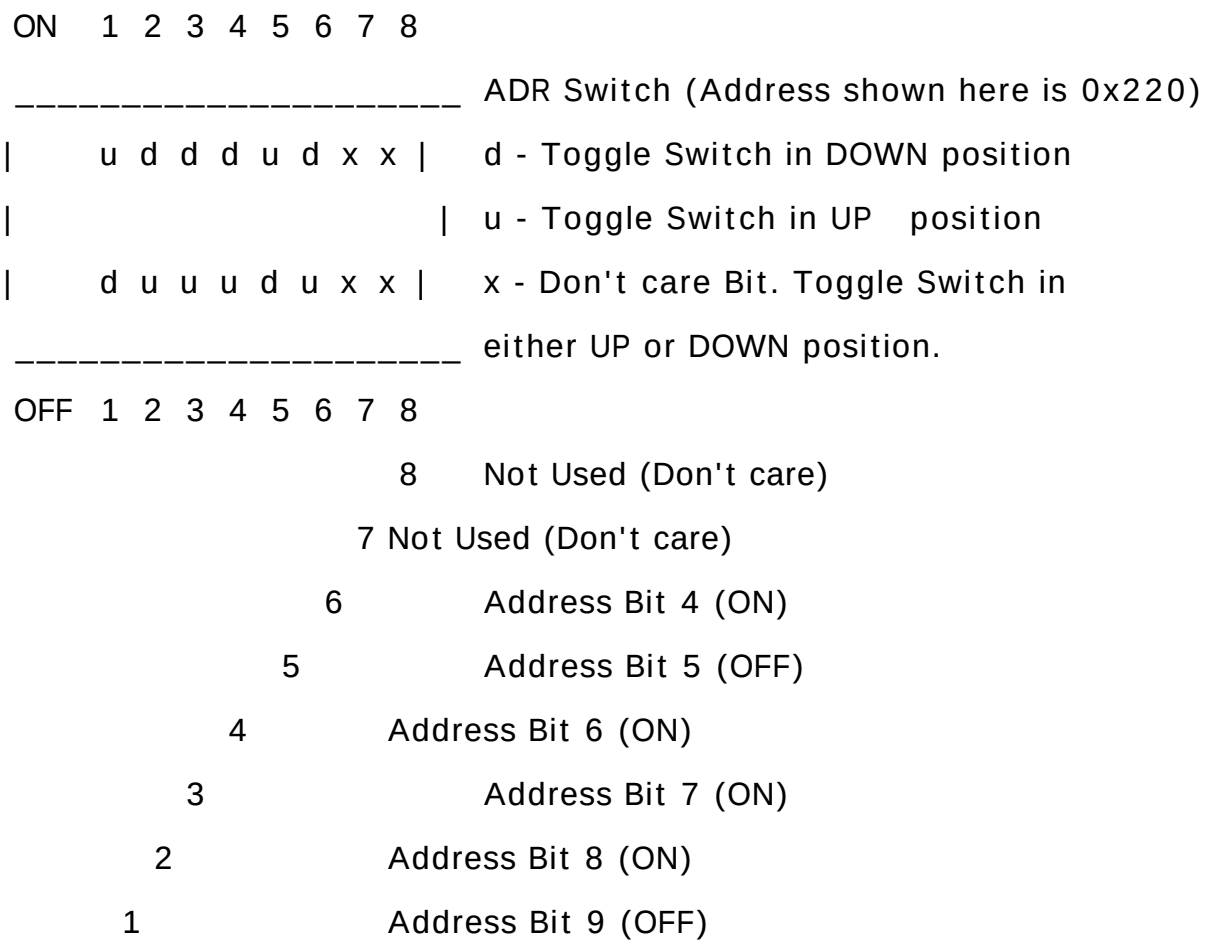
7.1 SCD NIB Hardware Configuration

Before a NIB card can be used it needs to be configured to the correct settings. This requires the Base Address switch for I/O address be set on the card. Each card's address must match the addresses used to install the NIB driver for each card installed in a system. The following table shows how to set the NIB I/O port address using the dip switches.

NIB I/O PORT ADDRESS SELECTION SWITCH

I/O Base Address Range(in Hex): 0x200 - 0x3f8

From Base Address the NIB card occupies 16 port addresses(e.g. 0x200-0x20f, 0x210-0x21f, 0x3a0-0x3af, etc.).



Each installed NIB card must have a unique base address. In general, the first ISA slot will have Serial I/F 0 set to address 0x220, the second ISA slot will have Serial I/F 1 set to 0x230, and the third ISA slot will have Serial I/F 2 set to 0x240. These boards/addresses correspond to the SCD operator interface screens that reference Serial I/F 0,1,and 2. Select the desired I/O base address from the table that follows. Use the above diagram as a guide for setting the toggle switches in correct position. The SCD software currently supports addresses 0x220 and 0x230 and will be upgraded to include 0x240 in release 3.0. Use only the addresses currently supported by the SCD software.

NIB BASE ADDRESS TABLE Settings 0x220-0x240

Address	1	2	3	4	5	6	7	8
0x220	OFF	ON	ON	ON	OFF	ON	X	X
0x230	OFF	ON	ON	ON	OFF	OFF	X	X
0x240	OFF	ON	ON	OFF	ON	ON	X	X

7.2 SCD NIB Software Configuration

To configure the NIB cards using the SCD operator interface, login as **scdadmin** and refer to Section 3.2.4.1 Modify Serial Interface Configuration Display in the Small Conversion Device Operator's Guide.

7.3 SCD NIB Verification

The installation of the NIB card can be validated in several ways.

During a system reboot, the NIB driver in the Linux kernel prints out a message with the IO address and assigned IRQ. Review the system startup messages using '**dmesg|more**' (Section 6.1). If the system boot messages do not correctly indicate the installed NIB cards, then the NIB cards have probably not been installed in the chassis, or have not had the dip switch set appropriately (Section 7.1).

If there is no startup message about NIBs at all, then it is possible that a Linux kernel is installed without support for NIBs. If so, this might require the re-installation of the 'vmlinuz' configuration item.

The SCD utility program **nibstatus** can be run from the scdtest account. The utility is currently configured to look for 4 NIB cards. It will report errors identifying 'missing' cards as /dev/nibS0, /dev/nibS1, /dev/nibS2, /dev/nibS3. Since most sites will have two NIB cards, there will normally be 2 error messages. If the nibstatus program displays statistics, then the nib cards indicated are installed and configured correctly and are accessible by the OS and application software.

If the nibstatus program indicates problems allocating /dev/nibS0, it is possible that the /dev directory does not include valid entries for the NIB cards. A directory listing of the /dev directory, '**ls -al /dev/nib***', should indicate entries in /dev for nibS0, nibR0, nibT0, nibS1, nibR1, nibT1 and they should all have attributes 'rw'. If not this will require re-installation of the 'linux' configuration item.

Section 8. SCD Software Updates

8.1 Introduction

The SCD application software is delivered on 3 1/2" diskettes and installed onto the SCD disk via the installation procedure described in Section 8.2. The SCD application software distribution floppy is labeled '**SCD_APP_SW'** on the printed label and internally on the diskette. For more details see Section 2 of the Small Conversion Device Operator's Guide.

8.2 Software and Configuration File Installation

The following menu is provided in the captive **scdcm** account to load the SCD software, load the SCD configuration, or create a SCD configuration backup (operator input is in bold type):

a. At the SCD login prompt, enter the userid and password

   scda login: **scdcm**

   password: **scdcm-password**

b. The following menu appears:

   SCD CM Menu

   1. Load SCD Software From Floppy
   2. Load SCD Configuration From Floppy
   3. Create SCD Configuration Floppy
   4. Logout

   Enter Selection:

c. To load the SCD software from floppy:

   Enter Selection: **1**

   When the SCD Software floppy disk has been inserted in drive,

   Hit Return key to continue: <**Return**>

   Loading SCD Software from floppy

   < list of all files copied from floppy >

   The SCD CM Menu will then be displayed.

d. To load the SCD configuration from floppy:

   Enter Selection: **2**

   When the SCD Configuration floppy disk has been inserted in drive,

   Hit Return key to continue: <**Return**>

   Loading SCD Configuration from floppy

   < list of all configuration files copied from floppy >

   The SCD CM Menu will then be displayed.

e. To create a SCD configuration floppy:

   Enter Selection: **3**

   When formatted floppy disk has been inserted in drive,

   Hit Return key to continue: <**Return**>

   Creating SCD Configuration on floppy

   < list of all configuration files copied to floppy >

   The SCD CM Menu will then be displayed.

f. To logout:

   Enter Selection: **4**

   Logging Out

   The SCD login prompt will then be displayed.

The new SCD software and/or configuration is now ready to be started by doing a coldstart or reboot.

Section 9. System Software Updates

9.1 Introduction

There are three means of delivering and applying updates to the SCD systems.

The SCD application software delivery procedure (Section 8.1) is designed and limited to delivering updates to SCD application code in directory /home/scdcm/scd.

A more general update procedure, described in Section 9.2, can apply updates to any system or user file on the system. This procedure is intended to provide emergency updates on diskette to the delivered systems as necessary. It can apply, for instance, updates to the password file or system startup procedures. The delivery diskettes are labeled internally and externally as SCD_SYS_UPD. The installer must have root userid access to apply these updates. Section 9.2.4 provides a means for remotely installing updates via the NOC subnet and the NOC ftp server.

A complete recovery procedure is also available to install, recover or upgrade the whole SCD system and software baseline by means of anonymous FTP from the central fileserver. This is described in Section 11.

9.2 System Software Update Installation

The installer must login as (or 'su' to) user root with the appropriate password.

An automated procedure 'sysupdate' is available, after the initial SCD release, to apply the updates.

9.2.1Applying System Update Diskette With Sysupdate

login (or su) to root

>**cd /root**

Insert the floppy labeled SCD_SYS_UPD in the floppy drive, and type:

>**sysupdate**

Apply SCD System Updates

Applying updates from diskette in floppy drive

Applying system updates from file 'directory/filename'

Do you want to apply these updates?

>**y**

System update applied successfully.

9.2.2 Applying System Updates Diskette Manually

The system updates must be applied from the root directory, with root access as follows:

>**whoami**

root

>**cd /**

>**pwd**

/

> **tar -V SCD_SYS_UPD -xzpf /dev/fd0**

9.2.3Applying System Update From The Fileserver With Sysupdate

Login (or su) to root.

A remote user must telnet to the SCD and login as scdinstl using the scdinstl password, then su to root using the root password.

Move to the /root directory.

>**cd /root**

FTP to the fileserver to retrieve the necessary system update file.

>**FTP 150.100.100.100 anonymous myname@**

>**cd pub**

>**bin**

>**get scd.sysupdate.tgz**

>**bye**

>**sysupdate scd.sysupdate.tgz**

Apply SCD System Updates

Applying updates from diskette in floppy drive

Applying system updates from file ‘directory/filename’

Do you want to apply these updates?

>**y**

System update applied successfully.

9.2.4Remotely Applying System Update Using The Fileserver

Login to one of the NOC machines (SCDs will only allow telnet sessions from IP addresses in the 150.144.180.40 subnet).

Telnet to device to be updated and login as scdinstl or scdtest.

Switch user to root:

>su - (use dash to force home directory to /root)

Type in password.

Check boot partition:

>df

Partition should be /dev/hda2. If it is /dev/hda3, reboot and try again. If it is still /dev/hda3, notify the system administrator. Continue with these procedures, but note that you will have to stop the update process earlier than normal.

Ensure that you are in the root directory:

>**pwd**

If you are not, change directories:

>cd /root

FTP to the fileserver to retrieve the necessary system update script file.

>**ftp 150.144.180.43 anonymous**

>**cd pub**

>**get update.sh**

>**bye**

This file is a command file which will ftp to the file server, fetch the necessary update files, and apply them to first the current partition (assumed to be /dev/hda2), and secondly to partition /dev/hda3.

Change the mode of the file so that it is executable and run it:

>chmod +x update.sh

>update.sh

It will step through the update process, asking you to press ENTER as needed. You will need to enter a password for the ftp of the files: any character string will do. If this machine could not be booted on /dev/hda2, enter **<ctrl> c** to stop the update process when it pauses just before updating the second partition (which would be /dev/hda3 again).

9.2.5Applying System Updates From The Fileserver Manually

Login (or su) to root.

A remote user must telnet to the SCD and login as scdinstl using the scdinstl password, then su to root using the root password.

Move to the /root directory.

>**cd /root**

FTP to the fileserver to retrieve the necessary system update file.

>**FTP 150.100.100.100 anonymous myname@**

>**cd pub**

>**bin**

>**get scd.sysupdate.tgz**

>**bye**

The system updates must be applied from the root directory, with root access as follows:

>**whoami**

root

>**cd /**

>**pwd**

/

> **tar -V SCD_SYS_UPD -xzpf /root/scd.sysupdate.tgz**

9.3 Distribution Files On The Fileserver

This section describes the SCD distribution files that are available on the fileserver.

The SCD system delivery is packaged as 5 separate software configuration items. If necessary, these packages can be retrieved from the fileserver and applied to re-create the correct SCD configuration.

This delivery file structure is designed to be used to recover a SCD after a disk crash or hard drive re-format. It also can be used to upgrade a SCD to a new base release, or to re-apply a miss-configured, missing, or deleted component from the standard SCD CM baseline.

The system configuration items and baseline filenames are listed below. The actual filenames on-line will include a release and version indication.

The suffix .tgz indicates a file that is created with a UNIX tar (Tape Archive) program, then compressed with the gzip utility. The Linux tar program (al la GNU) can read this tar-gzip format automatically when the 'z' attribute is specified.

Linux Baseline	scd.linux.tgz	The majority of the Linux system with executables, libraries and utilities. Directories: /bin, /boot, /dev, /lib, /proc, /root, /sbin, /tmp, /usr, /var
ETC Baseline	scd.etc.tgz	The system configuration and security files including system startup scripts and userids and passwords. Directory: /etc
Kernel Baseline	scd.vmlinuz.tgz	The Linuz system kernel with built-in device drivers for Ethernet cards and NIB cards. File: /vmlinuz
HOME Baseline	scd.home.tgz	The SCD application baseline directories and a version of the SCD application software. Directories: /home
Software Baseline		The SCD source code is not installed on-site.

In addition to these files reflecting the delivered configuration items, there are smaller filesets used for delivering software updates. These distributions are designed to be delivered by floppy or FTP.

SCD System Update	scd.sysupdate.tgz	An online system update file with updates to any of the system or application files. This file may also be delivered as a system update diskette labeled SCD_SYS_UPD.
SCD Appl. Software	scd.scdappsw.tgz	An online scd software delivery. This file may also be delivered as an application update diskette labeled SCD_APP_SW.

The SCD configuration files are not part of the CM baseline. They are unique to each SCD. The configuration file should be saved and restored to diskette using the scdcm menu, or reconfigured manually using the scdadmin menu as necessary.

The SCD emergency/rescue procedures (Section 11) describe how to install these files to re-create a whole scd system.

9.4 Distribution Media

The SCD system uses several diskettes for distribution. These are listed below.

Name	Label / Format	Description
SCD Application	SCD_APP_SW	A SCD application software delivery.
	Serial .tgz	A copy of the software release directory /home/scdcm/scd in tar-gzip format, copied to floppy. This tar file must be expanded in the /home/scdcm/scd directory
SCD System Update	SCD_SYS_UPD Serial .tgz	A copy of the system files to be delivered in tar-gzip format, copied to the floppy. This file must be expanded in the / directory.
Configuration file	CONFIG	A copy of the SCD configuration file.
	Serial	This diskette format can be loaded or saved from the scdcm menu.
Ethernet/Diagnostic	DIAGNOSTIC DOS Bootable	This diskette is used to diagnose certain PC problems. It is a DOS Bootable diskette. The AMIDIAG utility can be used to check system performance and settings. The 3C90XCFG utility can be used to checkout or configure the Ethernet cards.
Emergency BOOT	RESCUE Linux Bootable	This diskette is used to fix certain problems with the Linux file system. The booted OS provides sufficient capability to checkout the hard drive, reformat it, download a full SCD system and re-install it.

The same physical media, 3.5” high density diskettes, are used for Linux, DOS, and OS/2. The disks are formatted with either DOS ‘format’ or Linux ‘fdformat’.

The Linux diskettes cannot be verified under DOS since there is no DOS file system on them. Even though DOS reports that the disk is ‘unformatted’, don’t trash them!

Section 10. SCD Test Tools

10.1 SCD Test Tools

The SCD test tools are included in the SCD software delivery. The tools can be accessed by logging into the 'scdtest' account. A list (ls -al) of the currently available tools will be displayed.

- nibapp - transmits and receives data on a specified NIB card (similar to rd_nib and wr_nib)
- nibstatus - shows the status of the NIB cards
- rd_nib - reads serial blocks from the specified NIB card
- wr_nib - writes serial blocks to the specified NIB card
- view_net - reads everything sent to a specified multicast address/port
- zero_counts - will zero the serial and UDP block counts stored in shared memory
- mc_test - sends encapsulated test blocks to a specified multicast address/port
- show_counts - shows the serial and UDP block counts stored in shared memory
- show_dsites - shows the destination code routing table
- show_groups - shows the joined multicast groups
- show_procs - shows the process status
- show_sernum_out - shows the output serial block counts

The following sections describe the test tools by showing the available options, an example of the command line, and the output of the test tool. All of the show* tools display the same information as described in the SCD Operator's Guide.

10.2 NIBAPP

Executable: nibapp {options}

Options: -B(back2back default)

-F(fixed) usec

-P(playback) usec

-i device

-b baud or -z synthfreq

-t (timestamp)

-# copies

-m size

-v (validate)

-e (enum blocks)

-D hex_databyte

-c (crc encode/decode)

-h (enable hardware crc)

-s(sleep) sec

-V(printout version

-p(padbytes) nbytes

-o(synth osc) mhzfreq

-x 1 or 0 (tx external clk on/off)

-l logfile

Example: **nibapp**

Output: The output of nibapp(write) and nibapp(read) are shown in Figures 9-1 and 9-2.

```
drwxr-xr-x  13 scdcm    users        1024 Aug  8 18:23 ../
-rwxrwxr-x   1 scdcm    users       25774 Aug  8 09:23 mc_test*
-rwxr-xr-x   1 scdcm    users       20038 Jul 23 17:12 nibapp*
-rwxr-xr-x   1 scdcm    users       13075 Jul 23 17:12 nibstatus*
-rwxrwxr-x   1 scdcm    users       48980 Aug  8 09:24 rd_nib*
-rwxrwxr-x   1 scdcm    users       34455 Aug  8 09:24 show_counts*
-rwxrwxr-x   1 scdcm    users       34115 Aug  8 09:24 show_dsites*
-rwxrwxr-x   1 scdcm    users       34551 Aug  8 09:24 show_groups*
-rwxrwxr-x   1 scdcm    users       35826 Aug  8 09:24 show_procs*
-rwxrwxr-x   1 scdcm    users       33895 Aug  8 09:24 show_sernum_out*
-rwxrwxr-x   1 scdcm    users       24075 Aug  8 09:24 view_net*
-rwxrwxr-x   1 scdcm    users       88743 Aug  8 09:24 wr_nib*
-rwxrwxr-x   1 scdcm    users       35939 Aug  8 09:24 zero_counts*
scdd:~> nibapp -T
nibapp: illegal option -- T
usage: nibapp [-B(back2back default)] [-F(fixed) usec] [-P(playback) usec]\
[-i device] [-b baud || -z synthfreq] [-t (timestamp)] [-# copies] [-m size]\
[-v (validate)] [-e (enum blocks)] [-D hex_databyte] [-c (crc encode/decode)]\
[-h (enable hardware crc)] [-s(sleep) sec] [-V(printout version] \
[-p(padbytes) nbytes] [-o(synth osc) mhzfreq] \
[-x 1 or 0 (tx external clk on/off)] [-l logfile]
scdd:~> nibapp -i T0
/dev/nibT0 Blk no: 1679
scdd:~>
```

Figure 10-1. NIBAPP Write Output Screen

```
drwxr-xr-x  13 scdcm    users        1024 Aug  8 18:23 ../
-rwxrwxr-x   1 scdcm    users       25774 Aug  8 09:23 mc_test*
-rwxr-xr-x   1 scdcm    users       20038 Jul 23 17:12 nibapp*
-rwxr-xr-x   1 scdcm    users       13075 Jul 23 17:12 nibstatus*
-rwxrwxr-x   1 scdcm    users       48980 Aug  8 09:24 rd_nib*
-rwxrwxr-x   1 scdcm    users       34455 Aug  8 09:24 show_counts*
-rwxrwxr-x   1 scdcm    users       34115 Aug  8 09:24 show_dsites*
-rwxrwxr-x   1 scdcm    users       34551 Aug  8 09:24 show_groups*
-rwxrwxr-x   1 scdcm    users       35826 Aug  8 09:24 show_procs*
-rwxrwxr-x   1 scdcm    users       33895 Aug  8 09:24 show_sernum_out*
-rwxrwxr-x   1 scdcm    users       24075 Aug  8 09:24 view_net*
-rwxrwxr-x   1 scdcm    users       88743 Aug  8 09:24 wr_nib*
-rwxrwxr-x   1 scdcm    users       35939 Aug  8 09:24 zero_counts*
scdd:~> nibapp -R
nibapp: illegal option -- R
usage: nibapp [-B(back2back default)] [-F(fixed) usec] [-P(playback) usec]\
[-i device] [-b baud || -z synthfreq] [-t (timestamp)] [-# copies] [-m size]\
[-v (validate)] [-e (enum blocks)] [-D hex_databyte] [-c (crc encode/decode)]\
[-h (enable hardware crc)] [-s(sleep) sec] [-V(printout version] \
[-p(padbytes) nbytes] [-o(synth osc) mhzfreq] \
[-x 1 or 0 (tx external clk on/off)] [-l logfile]
scdd:~> nibapp -i R0
/dev/nibR0 Blk no: 578
scdd:~>
```

Figure 10-2. NIBAPP Read Output Screen

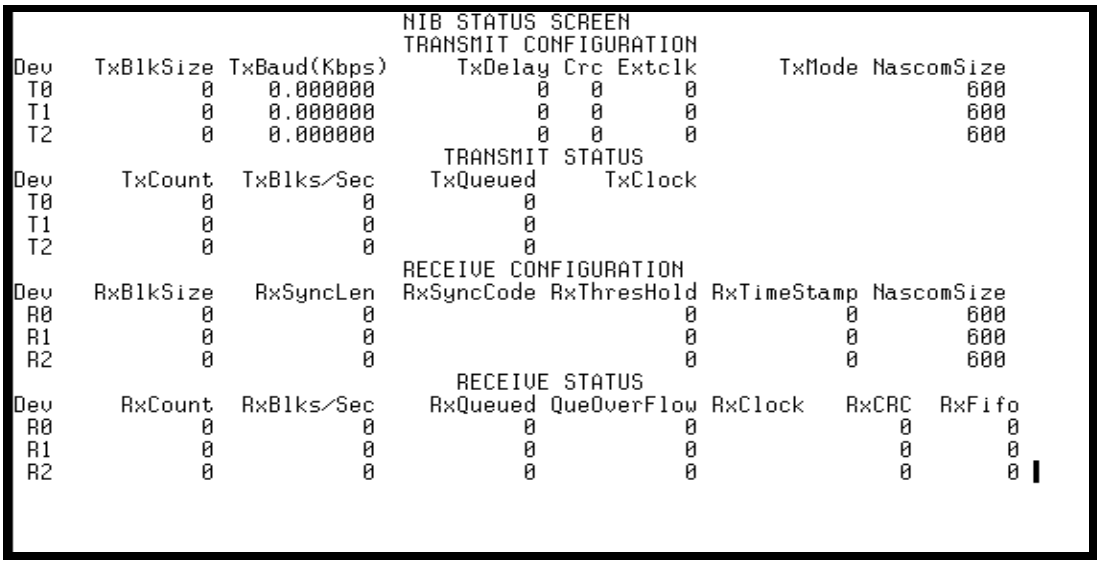
10.3 NIBSTATUS

Executable: nibstatus

Options: none

Example: **nibstatus**

Output: The output of nibstatus is shown in Figure 10-3.



```
                          NIB STATUS SCREEN
                          TRANSMIT CONFIGURATION
Dev   TxBlkSize TxBaud(Kbps)     TxDelay Crc Extclk      TxMode NascomSize
 T0           0    0.000000           0   0      0                     600
 T1           0    0.000000           0   0      0                     600
 T2           0    0.000000           0   0      0                     600
                           TRANSMIT STATUS
Dev     TxCount  TxBlks/Sec    TxQueued     TxClock
 T0           0           0           0
 T1           0           0           0
 T2           0           0           0
                          RECEIVE CONFIGURATION
Dev   RxBlkSize   RxSyncLen  RxSyncCode RxThresHold RxTimeStamp NascomSize
 R0           0           0                       0           0        600
 R1           0           0                       0           0        600
 R2           0           0                       0           0        600
                             RECEIVE STATUS
Dev     RxCount  RxBlks/Sec    RxQueued QueOverFlow RxClock   RxCRC  RxFifo
 R0           0           0           0           0                0       0
 R1           0           0           0           0                0       0
 R2           0           0           0           0                0       0
```

Figure 10-3. NIBSTATUS Output Screen

10.4 RD_NIB

Executable: rd_nib {options}

Options: -b src/dest/type/fill --> block attributes

-d device_name --> /dev/nibR0 - /dev/nibR2

-h --> help

-N --> turn off sequence numbering

-n number_blks --> stop after receiving number

-p --> check poly code

-s number_blks --> show status after X blocks

-S --> print sequence error info

-v --> turn off all validation

-V --> turn off block data checking

-Y --> extra verbose stuff

Example: Listen on NIB device /dev/nibR1 for blocks. Validate the following: source = 10, destination = 20, type = 15,

fill pattern = 0xab. Report a status back the screen every 100 blocks.

rd_nib -d /dev/nibR1 -b 10/20/15/ab -s 100

Output: The output of rd_nib is shown in Figure 10-4.

```
scdg:~> rd_nib -d /dev/nibR0 -b 10/20/15/ab -p -s 100
Receiving Nascom blocks:
        Source Code 10, Data Type code 15, and starting Dest Code 20
        Block fill pattern is ab
        -       Polynomial code generation verification enabled
        -       Sequence count verification enabled
        -       Block data verification enabled
rd_nib src 10, dest 20, blocks 100, errors 0/0/0, seq_err 0
rd_nib src 10, dest 20, blocks 100, errors 0/0/0, seq_err 0
Destination Code Distribution
        0       1       2       3       4       5       6       7       8       9
016:   000100 000000 000000 000000 000000 000000 000000 000000 000000 000000
026:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
036:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
046:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
056:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
066:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
076:   000000 000000 000000 000000
Sequence Error Distribution
        0       1       2       3       4       5       6       7       8       9
016:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
026:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
036:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
046:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
056:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
066:   000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
076:   000000 000000 000000 000000
scdg:~>
```

Figure 10-4. RD_NIB Output Screen

10.5 WR_NIB

Executable: wr_nib {options}

Options: -b src/dest/type/fill --> block attributes

-d device_name --> /dev/nibT0 - /dev/nibT2

-h --> help

-L range_num --> generate linear dest codes

-n num_of_blocks --> number of blocks to write

-N --> turn off sequence numbering

-p --> generate poly code

-P number_bytes --> number of Pad bytes

-r baud_rate --> baud rate in khz .02 - 1544.0

-R range_num --> generate random dest codes

-s frequency --> write summary to every X blocks

-x --> use external clock

-Y --> extra verbose stuff

Example: Transmit 10000 blocks on NIB device /dev/nibT1.

Initialize the blocks as: source = 10, destination = 20, type = 15,

fill pattern = 0xab. Report a status back the screen every 100 blocks.

Use a data rate of 1024 khz.

wr_nib -d /dev/nibT1 -b 10/20/15/ab -s 100 -n 10000 -r 1024

Output: The output of wr_nib is shown in Figure 10-5.

```
Hardware poly code generation enabled
        -       Sequence count generation enabled
        -       Current clock rate is 448000 hz
Hit return to terminate WRITE_BOARD -->
wr_nib wrote block 100, src 10, dest 20, type 15
scdg:~> wr_nib -d /dev/nibT0 -b 10/20/15/ab -p -s 100 -n 1000 -r 448
Generating Nascom blocks:
        Source Code 10, Data Type code 15, and starting Dest Code 20
        Block fill pattern is ab
Hardware poly code generation enabled
        -       Sequence count generation enabled
        -       Current clock rate is 448000 hz
wr_nib wrote block 100, src 10, dest 20, type 15
wr_nib wrote block 200, src 10, dest 20, type 15
wr_nib wrote block 300, src 10, dest 20, type 15
wr_nib wrote block 400, src 10, dest 20, type 15
wr_nib wrote block 500, src 10, dest 20, type 15
wr_nib wrote block 600, src 10, dest 20, type 15
wr_nib wrote block 700, src 10, dest 20, type 15
wr_nib wrote block 800, src 10, dest 20, type 15
wr_nib wrote block 900, src 10, dest 20, type 15
Hit return to terminate WRITE_BOARD -->
wr_nib wrote block 1000, src 10, dest 20, type 15
scdg:~>
```

Figure 10-5. WR_NIB Output Screen

10.6 VIEW_NET

Executable: view_net {options}

Options: -a net address --> net address to listen

-d num --> dump num bytes in hex

-D nascom_dest --> filter using NASCOM dest

-f hex_byte --> hex fill pattern

-h --> help

-I --> use interactive viewing

-n num_blks --> stop after num_blks

-p port_num --> port_num of listen

-r range --> listen group range

-s source_addr --> filter using source address

-S nascom_source --> filter using NASCOM source

-v num_blks --> turn on validation and report after num_blks

-Y --> extra verbose stuff

Example: Listen for multicast group 224.0.1.20 port 8001

in interactive mode and dump 40 bytes of the message.

Show me only blocks with Nascom destination code 12.

view_net -p 8001 -a 224.0.1.20 -i -d 40 -D 012

Output: The output of view_net is shown in Figure 10-6.

```
scdd:~> view_net -a 225.168.201.113 -p 9050
Receiving Nascom Encapsulated blocks:
        view_net is in view only mode.
Joined group 225.168.201.113
Source Ip 192.168.201.113, 1, Seq number 0 Source 0 Dest 4
Listen Ip 225.168.201.113 Port 9050
Total Number Blocks Read 1
Total Errors Data 0, Sequence 0, Sync 0
scdd:~>
scdd:~>
scdd:~>
scdd:~>
scdd:~>
scdd:~>
```

Figure 10-6. VIEW_NET Output Screen

10.7 ZERO_COUNTS

Executable: zero_counts

Options: none

Example: **zero_counts**

Output: There is no output of zero_counts. This will zero out the UDP and serial input and output counts stored in shared memory by the SCNTR process.

10.8 MC_TEST

Executable: mc_test {options}

Options: -a net address --> net address for broadcast

-f src/dest/type/fill --> Nascom block fill pattern

-h --> help

-n num_blks --> stop after num_blks

-p port_num --> port_num of broadcast

-r data_rate --> transmit data rate

Example: Transmit 10000 blocks to multicast group 224.0.1.20 port 8001

at 1024 kbps. Fill the block with the following Nascom

block info: source = 10 ,destination = 20 , type = 15 and

data fill pattern of 0xab.

mc_test -a 224.0.1.20 -p 8001 -r 1024 -f 10/20/15/ab -n 10000

Output: The output of mc_test is shown in Figure 10-7.

```
scdd:~>
scdd:~>
scdd:~>
scdd:~>
scdd:~>
scdd:~> mc_test -a 225.168.201.113 -p 9050 -r 1024 -f 10/20/15/ab -n 10
We'll send about 213.333328 blocks for 0.046875 seconds
Actually send 5.000000 blocks and wait 23438 microseconds
DONE!!! Sent 10 blocks final seq_num = 10
scdd:~>
scdd:~>
scdd:~>
scdd:~>
scdd:~>
```

Figure 10-7. MC_TEST Output Screen

10.9 SHOW_COUNTS

Executable: show_counts

Options: none

Example: **show_counts**

Output: The output of show_counts is shown in Figure 10-8.

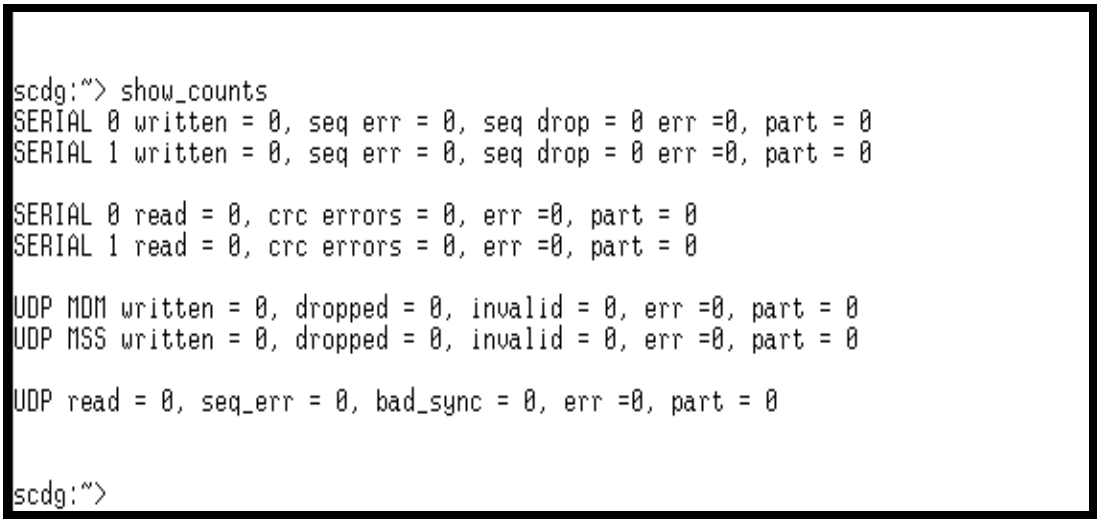


Figure 10-8. SHOW_COUNTS Output Screen

The first two lines pertain to serial output. The line beginning SERIAL 0 pertains to board 0 and the line beginning SERIAL 1 applies to board 1. The value following ‘written =‘ is the number of successful calls made to the serial NIB driver with all data bytes transferred. This number corresponds to the value shown on the SCD Detail Counts display as the S-Xmit count for a particular Serial-IF. The two values for ‘err =‘ and ‘part =‘ do not appear on any SCD display. The value following ‘err =‘ is the number of times a call was made to the serial NIB driver and an error was returned . The value ‘part =‘ is the number of successful calls made to the serial NIB driver with less then the total number of data bytes transferred. The values discussed so far are accumulated by the serial output processes. The last two values are accumulated by the UDP input process, these two values are not displayed on any SCD display. The value following ‘seq err =‘ is the number of sequence errors detected for blocks destined to a particular serial NIB board. The value following ‘seq drop =‘ is the number of blocks dropped due to sequence errors when the configuration of the NIB board indicated that sequence errors should be blocked. This configuration item can be seen on the SCD Serial Interface Configuration display where the field Seq Err was set to drop rather than pass.

The third and fourth lines pertain to serial input. The line beginning SERIAL 0 pertains to board 0 and the line beginning SERIAL 1 applies to board 1. The value following ‘read =‘ is the number of successful calls made to the serial NIB driver with all data bytes requested being received. This number corresponds to the value shown on the SCD Detail Counts display as the S-Recv count for a particular Serial-IF. The two values for ‘err =‘ and ‘part =‘ do not appear on any SCD display. The value following ‘err =‘ is the number of times a call was made to the serial NIB driver and an error was returned . The value ‘part =‘ is the number of successful calls made to the serial NIB driver with less then the total number of data bytes requested being received. The value following ‘crc errors =‘ is the number of successful calls made to the serial NIB driver with all data bytes requested being received and the NIB

driver returned an additional status that the CRC check for the block had failed. This number corresponds to the value shown on the SCD Detail Counts display as the CRC Err count for a particular Serial-IF.

The fifth and sixth lines pertain to UDP output. The line beginning UDP MDM pertains to all UDP output where a NASCOM block was received on a serial NIB board configured for fixed routing while the line beginning UDP MSS pertains to all UDP output where a NASCOM block was received on a serial NIB board configured for table routing. The value following ‘written =‘ is the number of successful calls made to send data. This number corresponds to the value shown on the SCD Detail Counts display as the IP-Xmit count for a particular routing method. The two values for ‘err =‘ and ‘part =‘ do not appear on any SCD display. The value following ‘err =‘ is the number of times a call was made to send UDP data and an error was returned . The value ‘part =‘ is the number of successful calls made to send UDP data with less then the total number of data bytes transferred. The value following ‘dropped = ‘ is the number of blocks dropped while an update to the routing addresses is in progress. This number corresponds to the value shown on the SCD Detail Counts display as the Drop Blks count for a particular routing method. The value following ‘invalid =‘ is the number of blocks dropped because no address has been given for a particular destination code in the routing table. This number corresponds to the value shown on the SCD Detail Counts display as the Inv Dest count for a particular routing method. This value is meaningful for table routing only, the value for fixed routing will always be zero.

The seventh and final line pertains to UDP input. The value following ‘read =‘ is the number of successful calls made to receive UDP data with all data bytes requested being received. This number corresponds to the value shown on the SCD Detail Counts display as the IP-Recv count. The two values for ‘err =‘ and ‘part =‘ do not appear on any SCD display. The value following ‘rr =‘ is the number of times a call was made to receive UDP data and an error was returned . The value ‘part =‘ is the number of successful calls made to received UDP data with less then the total number of data bytes requested being received. The value following ‘bad_sync = ‘ is the number of full blocks received where the NASCOM sync pattern did not appear in the expected location. This number corresponds to the value shown on the SCD Detail Counts display as the Bad Sync count. The number following ‘seq err =‘ is not used and should always be zero.

10.10 SHOW_DSITES

Executable: show_dsites

Options: none

Example: **show_dsites**

Output: The output of show_dsites is shown in Figure 10-9.

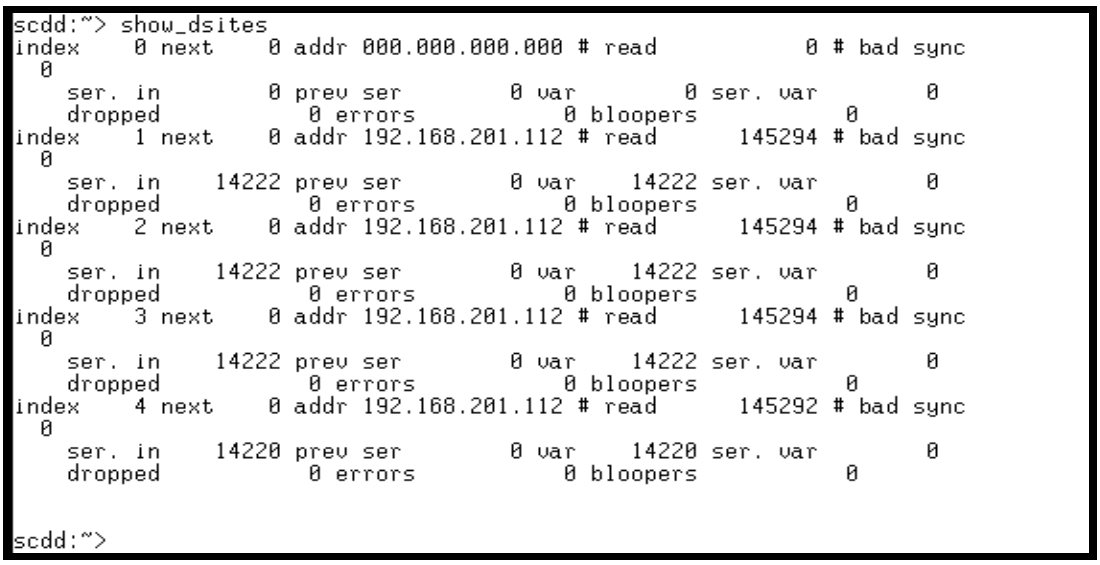


Figure 10-9. SHOW_DSITES Output Screen

The information from this tool expands the information show on the SCD Sequence Error Counts display. The tool show_dsites is a line by line entry format while show_groups shows the same information listed in tabular form. The show_dsite listing has a line entry for each Sender Address that has sent this SCD a block, however the listing does not associate the Multicast Address with the sender address. This has to be done manually by comparing the show_dsites entries with those shown on the SCD Sequence Error Counts display .

The 10 columns of information are also given in the show_groups list, the named value will be given in brackets. The 10 columns are:

1. the Senders IP address [addr] this is displayed on the Sequence Error Counts display as Sender Address
2. number of good blocks read [# read]
3. number of good blocks read where the NASCOM sync could not be located [# bad sync]
4. next expected sequence number [ser. in] this is displayed on the Sequence Error Counts display as Current Seq Num
5. not used, should always be zero [prev ser]
6. the next expected sequence number to be used in calculating the variance [var]
7. the calculated variance* [ser. var]
8. not used, should always be zero [dropped]
9. number of times the sequence number contained in the incoming UDP block differed from the expected sequence number [errors] this is displayed on the Sequence Error Counts display as Sequence Errors

10. number of times the sequence number contained in the incoming UDP block differed from the expected sequence number by more than 100 [bloopers] this is displayed on the Sequence Error Counts display as Sequence Sync Err

* As the Sequence Error Count goes up, the variance may indicate the type of sequence errors that are occurring. A positive value indicates blocks are being dropped, a negative number indicates duplicate blocks, and if the variance remains steady then either 1)the blocks are out of sequence or 2) dropped blocks and duplicate blocks are both occurring thus canceling each other out.

10.11 SHOW_GROUPS

Executable: show_groups

Options: none

Example: **show_groups**

Output: The output of show_groups is shown in Figure 10-10.

```
scdc:~> show_groups
    0    8  1  225.000.001.001   8001   1   0
  192.168.201.110    3000000          0  50880      0  50880      0      0
0       0
    1    9  1  225.000.001.002   8002   1   0
  192.168.201.110    2999999          0  50880      0  50880      1      0
1       0
    2   10  1  225.000.001.003   8003   1   0
  192.168.201.110    2999999          0  50880      0  50880      1      0
1       0
    3   11  1  225.000.001.004   8004   1   0
  192.168.201.110    2999999          0  50880      0  50880      1      0
1       0
    4   12  1  225.000.001.005   8005   1   0
  192.168.201.110    2999999          0  50880      0  50880      1      0
1       0
5 group entries used

scdc:~>
scdc:~>
scdc:~>
scdc:~>
scdc:~>
scdc:~>
scdc:~>
```

Figure 10-10. SHOW_GROUPS Output Screen

The information from this tool expands the information show on the SCD Sequence Error Counts display. The tool show_groups is in tabular format. The show_groups listing gives a line for the Multicast Address being listened to followed by a list of all Sender Addresses from which data has been received , a line of information for each sender. The show_groups listing will be described with references back to the named values in the show_dsites listing.

The show_groups table has two types of lines, one line for the Multicast Address information (has 7 columns) followed by one or more lines of Sender Address information (has 10

columns). The show_dsites listing does not contain the Multicast Address information. The 7 columns in the Multicast Address information are:

1. index into the group table
2. the socket descriptor
3. the entry state (1-in use, 2- being added, 3-being deleted, 4-reserved, 5-closing*)
4. listening (multicast) address this is displayed on the Sequence Error Counts display as Multicast Address
5. associated port address
6. send data out NIB board 0? (1->yes; 0->no)
7. send data out NIB board 1? (1->yes; 0->no)

* when closing is complete the entry state will be reserved but no socket will be opened, reserved means that the listening group is associated with a NIB board that is disabled and no NIB board that is enabled.

The 10 columns of information are also given in the show_dsites list, the named value will be given in brackets. The 10 columns are:

1. the Senders IP address [addr] this is displayed on the Sequence Error Counts display as Sender Address
2. number of good blocks read [# read]
3. number of good blocks read where the NASCOM sync could not be located [# bad sync]
4. next expected sequence number [ser. in] this is displayed on the Sequence Error Counts display as Current Seq Num
5. not used, should always be zero [prev ser]
6. the next expected sequence number to be used in calculating the variance [var]
7. the calculated variance* [ser. var]
8. not used, should always be zero [dropped]
9. number of times the sequence number contained in the incoming UDP block differed from the expected sequence number [errors] this is displayed on the Sequence Error Counts display as Sequence Errors
10. number of times the sequence number contained in the incoming UDP block differed from the expected sequence number by more than 100 [bloopers] this is displayed on the Sequence Error Counts display as Sequence Sync Err

* As the Sequence Error Count goes up, the variance may indicate the type of sequence errors that are occurring. A positive value indicates blocks are being dropped, a negative number

indicates duplicate blocks, and if the variance remains steady then either 1)the blocks are out of sequence or 2) dropped blocks and duplicate blocks are both occurring thus canceling each other out.

10.12 SHOW_PROCS

Executable: show_procs

Options: none

Example: **show_procs**

Output: The output of show_procs is shown in Figure 10-11.

```
scdd:~> show_procs
  i    name    pid  type  state   mode   num
  0    scntr    83     1     1     2    0
  1    ssero     0    10     0     1    0
data_rate = 1544000 rate_id = 202 external = 0 time delay = 0
mfactor = 134231070  synth_osc = 20000000
seq number action = 1 cab action = 0 test gen. = 0
gen poly flag = 0
  2    ssero     0    11     0     1    0
data_rate = 0 rate_id = 0 external = 0 time delay = 0
mfactor = 0  synth_osc = 0
seq number action = 0 cab action = 0 test gen. = 0
gen poly flag = 0
  3    sudpo     0     3     0     0    0
MDM TTL value = 0
MDM board 0 xmit_cab = 0 cab_rate = 0 time_tag = 0
MDM board 1 xmit_cab = 0 cab_rate = 0 time_tag = 0
MDM board 0 ip addr = 00000000 and ip port 0  serial num = 0
MDM board 1 ip addr = 00000000 and ip port 0  serial num = 0
MDM board 0 dropped = 0 sernum out 0
MDM board 1 dropped = 0 sernum out 0
  4    sudpo     0     4     0     1    0
MSS TTL value = 10
  5    sseri     0    20     0     1    0
  6    sseri     0    21     0     1    0
  7    sudpi     0     2     0     2    0
  8     scfg     0     5     0     2    0
```

Figure 10-11. SHOW_PROCS Output Screen

The layout is an internal process table (7 columns) and some of the information can be viewed on the SCD Process Status display. For some processes, extended information is given. The seven columns are:

1. entry position within the process table
2. the name of the process
3. the UNIX Process ID (0 means not active)

4. .type of process (1-> SCD control, 2-> udp input, 3->fixed routing udp output, 4 ->table routing udp output, 5 ->udate configuration, 10->serial output board 0, 11-> serial output board 1, 20 -> serial input board 0, 21->serial input board 1)
5. process state (0 -> down, 1->up, 2->terminating, 3->starting, 4->disabled, 5->restarting)
6. process mode (0-> fixed routing, 1-> table routing, 2-> not applicable)
7. the number of times an attempt has been made to restart the process

The serial output processes (ssero) have additional values, some of which can be seen on the SCD Serial Interface Configuration display. These values are:

- data_rate: data rate at which the board is enabled
- rate_id: value passed to NIB driver specifying the data rate
- external: 0-> internal clock, 1->external clock
- time_delay: 0-28 number of fill bytes to put at the end of each serial block
- mfactor: value passed to NIB driver for synthesized clocking
- synth_osc: rate of synthesizer oscillator
- seq number action (0->drop blocks out of sequence, 1-> pass out of sequence blocks on)
- cab action (0->drop CAB blocks, 1-> pass CAB blocks on)
- test_gen: not used, always zero
- gen poly flag (0->don't generate poly, 1-> generate poly)

The UDP output process for fixed routing (MDM) have additional values, some of which can be seen on the SCD Serial Interface Configuration display. All but the TTL value have entries for both NIB board 0 and 1. These values are:

- TTL value: Multicast time to live value
- xmit_cab: (0->no CABS generated, 1->CABS generated)
- cab_rate: rate (milliseconds) at which CABS are generated on an idle line
- time_tag: not used, always zero
- ip addr: the address to which data will be sent for fixed routing
- ip port: the ip port associated with the ip addr for fixed routing
- serial_num: same as sernum_out
- dropped: the number of blocks dropped because the ip addr or ip port was being changed'
- sernum out: the next fixed routing sequence number that will be used

The UDP output process for fixed routing (MSS) has the additional TTL value representing the multicast time to live.

10.13 SHOW_SERNUM_OUT

Executable: show_sernum_out

Options: none

Example: **show_sernum_out**

Output: The output of show_sernum_out is shown in Figure 10-12.

```
scdd:~> show_sernum_out ! more
 dest     ip address       port  serial number
   1  225.000.001.001     8001    000000
   2  225.000.001.002     8002    000000
   3  225.000.001.003     8003    000000
   4  225.000.001.004     8004    000000
   5  225.000.001.005     8005    000000
   6  225.000.001.001     8001    000000
   7  225.000.001.001     8001    000000
   8  225.000.001.001     8001    000000
   9  225.000.001.001     8001    000000
  10  225.000.001.001     8001    000000
  11  225.000.001.001     8001    000000
  12  225.000.001.001     8001    000000
  13  225.000.001.001     8001    000000
  14  225.000.001.001     8001    000000
  15  225.000.001.001     8001    000000
  16  225.000.001.001     8001    000000
  17  225.000.001.001     8001    000000
  18  225.000.001.001     8001    000000
  19  225.000.001.001     8001    000000
  20  225.000.001.001     8001    000000
  21  225.000.001.001     8001    000000
  22  225.000.001.001     8001    000000
  23  225.000.001.001     8001    000000
```

Figure 10-12. SHOW_SERNUM_OUT Output Screen

This tool my be used to display the next sequence number to be used when serial data is encapsulated with an RTP header and sent UDP out on the ethernet. This tabular table applies to table routing only. There are four columns which represent the NASCOM Destination Code found in the 4800 bit block, the IP address that the block will be sent . the IP port , and the next sequence number to be used.

Section 11. Disk Drive Rescue

The following procedures may be used to completely install, recover and/or upgrade an entire SCD disk. This procedure is designed to be used when the SCD cannot be booted from the hard disk drive and a "rescue" floppy diskette must be used to boot.

The rescue boot floppy can also be used as a diagnostic diskette, to boot the system to run a file system check on the hard drive.

11.1 Boot Rescue System from Floppy

1. Insert Boot floppy and depress hardware reset button, or power on.
2. Depress the DEL key during bootup to go into BIOS CMOS Setup
3. Tab to the 'advanced' menu and change the 'BootUp Sequence' to 'A:C:CDROM'
   so the system will boot from the floppy drive.
   Remember that this BIOS setting must be set back to 'C:A:CDROM' when the SCD is in use. If not, the SCD coldstart procedures will hang when there is a SCD Configuration floppy that in the disk drive.
4. Use ESCape to leave BIOS setup and save the settings.
5. Answer the following IP (netconfig) prompts:
6. Do you plan to ONLY use loopback ([y]es, [n]o)? **n**
7. Enter your IP address for the local machine. Example: 111.112.113.114
8. Enter IP address for SCD_RESCUE (aaa.bbb.ccc.ddd): **192.168.201.114**
9. Enter gateway address (aaa.bbb.ccc.ddd): **192.168.201.161**
10. Enter netmask (aaa.bbb.ccc.ddd): **255.255.255.0**
11. Will you be accessing a nameserver ([y]es, [n]o)? **n**
12. Remove floppy diskette from drive.
13. Login as root

11.2 Set Up Hard Drive Partition Table with FDISK

The following instructions are for the Linux version of fdisk.

The Linux disk partitititions are compatable with DOS and OS/2. Either of these systems fdisk utilities may be used to allocate the partitions instead of the Linux version of FDISK. If so, then the Linux fdisk utility is only necessary to set the partition type for the Linux swap and Linux native partitions.

The allocation of the disk partitions is fairly flexable. The standard configuration is to allocate a 64 Mbyte swap area and two redundant partitions for the SCD system.

1. Type **fdisk** to start the program
2. Select option **m** for help menu.

3. Allocate (option **n**) a 64 Mbyte linux swap area.
   64 Mbytes = 66 cyls of 2048*512 bytes = 66,496 blocks
4. Set swap partition type (option **t**) to Linux swap (**82**)

5. Allocate (option **n**) a Linux Native partition of about half the available space.
   410 Mbytes = 410,256 blocks of 2048*512

6. Set Linux partition (option **t**)to Linux native (**83**).

7. Allocate (option **n**) remainder to Linux Native.
   356 Mbytes = 356,832 blocks of 2048*512
8. Set Linux partition (option **t**)to Linux native (**83**).

9. List partitions (option p) and write them down for later:
   hda1 swap
   hda2 Linux Native
   hda3 Linux Native (alternate boot partition)
10. Write partition to disk (option **w**)
11. Quit program (option **q**)

11.3 Create File Systems

The '-c' option on these utilities will invoke a read integrity check of the filesystem. It takes a bit longer but may be worthwhile if the hard disk is suspect. It may be omitted if time is critical and the risk of disk error is considered acceptable.

1. Create swap area on the 65K partition defined above:
   mkswap -c /dev/hda1
2. Create Unix extended filesystem on large partition defined above:
   mke2fs -c /dev/hda2
3. Create Unix extended filesystem on extra large partition defined above:
   mke2fs -c /dev/hda3

11.4 Mount Linux File System

1. **mount /dev/hda2 /mnt**
2. **cd /mnt**

At this time the root directory / refers to a memory resident copy of the root floppy disk. Files changed there will not be retained when you reboot. Directory /mnt is the first directory on the hard drive and will be the root directory / after booting to that partition on the hard drive.

11.5 Use FTP to Download System Tar Files

The whole SCD system is available on an anonymous FTP fileserver. The system is delivered as 4 configuration items saved as Unix tar files, compressed with GZIP.

1. Get to top of hard drive file system:
   cd /mnt
2. **ftp 150.144.180.45** (valid address of fileserver)
3. userid: **anonymous**
4. password: ***sitename***
5. get to distribution directory: **cd pub**
6. List the directory of available system update versions: **ls -l**

   There may be multiple versions of the delivery files available if various releases are deployed in the field. Please note the highest applicable release filename for the linux, home, etc and vmlinux files.

7. get scd.linux.tgz
8. get scd.home.tgz
9. get scd.etc.tgz
10. get scd.vmlinuz.tgz
11. bye

11.6 Expand System Tar Files

1. Make sure you are still in root of hard disk:
   cd /mnt
2. **tar xzpf scd.linux.tgz**
3. **tar xzpf scd.home.tgz**
4. **tar xzpf scd.etc.tgz**
5. **tar xzpf scd.vmlinuz.tgz**

If there is an error while expanding the files, it is possible that the downloaded tar file is corrupted on the local hard drive. The scd.linux.tgz file in particular is very large and is likely to incur errors if the hard drive is not stable. The integrity of the downloaded files may be tested by means of '**GZIP -t scd.linux.tgz'** If GZIP indicates a checksum error, then attempt to download the file again, or replace the hard drive.

11.7 Check etc/fstab Entries

Set correct partitions hda1 or hda2 for root and swap filesystems. The SCD system will not boot (or not correctly) if these fstab settings disagree with disk partition they reside on.

The /etc/fstab entry on hda2 must indicate /dev/hda2 as root.

The /etc/fstab entry on hda3 must indicate /dev/hda3 as root.

1. **cd /mnt/etc**
2. **vi fstab**
   Check the entries
3. /dev/hda**2** / ext2 defaults 1 1
4. /dev/hda**1** swap swap defaults 1 1
5. **<esc>:wq** to write and quit the vi editor

11.8 Configure IP Address with Netconfig

The netconfig procedure is used to modify several files in the /etc directory that indicate the hostname, domain name and start the Ethernet device with the correct IP address.

Utilities like lilo and netconf will access and modify files in /etc. Rather than modify these procedures to access files in /mnt/etc it is simpler to use chroot.

1. Move the logical file system root to the top of the hard drive:
   chroot /mnt bash
2. Configure the IP on the harddrive:
   netconfig
   enter hostname, IP address, netmask, router address when prompted.

3. **exit** the chroot shell

11.9 Install System on the Alternate Linux Partition

It is convenient to maintain a duplicate filesystem on an alternate partition. This allows a fast recovery if the primary filesystem is corrupted, for instance, after a power failure. It also could be used to test an alternate release of the system or application software.

1. Mount the alternate filesystem.
2. cd /
3. mount /dev/hda3 /mnt/mnt
4. Check it
   df

   Should result in something like:
   filesystem 1024-blocks Used Available Capacity Mounted on
   /dev/hda2 961923 818377 93852 90% **/mnt**
   /dev/hda3 961923 818377 93852 90% **/mnt/mnt**
5.
6. cd /mnt
7. Copy the whole physical hda2 (on /mnt) partition to hda3 (on /mnt/mnt)
   cp -dpRx . mnt

Set the /etc/fstab entry correctly on the hda3 (/mnt/mnt) filesystem.

1. **cd /mnt/mnt/etc**
2. **vi fstab**
   Check the entries
3. /dev/hda**3** / ext2 defaults 1 1
4. /dev/hda**1** swap swap defaults 1 1
5. **<esc>:wq** to write and quit the vi editor

11.10 Set up Lilo for Booting

1. If configuring two partitions use df to check that both are mounted at this time
   df
   filesystem 1024-blocks Used Available Capacity Mounted on
   /dev/hda2 961923 818377 93852 90% **/mnt**
   /dev/hda3 961923 818377 93852 90% **/mnt/mnt**
2. Move the logical file system root to the hard drive:
   chroot /mnt bash

   For now, directory / refers to the root filesystem on the hard drive (under /mnt).
   Utilities like lilo and netconf automatically access or modify files in /etc.
   Rather than modify these procedures to access /mnt/etc, it is simpler to use chroot ('change root'). Note, though, that system utilities like ps, df or reboot will not work correctly in the chroot environment.

3. vi /etc/lilo.conf

4. Check /etc/lilo.conf for correct root partition

```
#   Lilo configuration for linux booting from two partitions
#   Modifies MBR on hda
boot = /dev/hda        # modifies this disks boot record
vga = normal   # force sane state
delay = 50             # waits this number of tenth seconds when booting
ramdisk = 0            # paranoia setting no ramdisk
read-only              # mound fs readonly for checkout
# End LILO global section
# Linux bootable partition config begins
# this section for default boot
image = /vmlinuz
  root = /dev/hda2
  label = linux-hda2
# This section for alternate boot to second partition
image = /mnt/vmlinuz
  root = /dev/hda3
  label = linux-hda3
# Linux bootable partition config ends

# Linux bootable partition config ends
```

This is an alternate lilo configuration as an example for use under OS2 bootmanager. It does not modify the disk's master boot area.

```
#   Leaves disks MBA alone !
install = /boot/boot.b# saves secondary loader here
vga = normal   # force sane state
ramdisk = 0            # paranoia setting
read-only              # mound fs readonly for checkout
# End LILO global section
# Linux bootable partition config begins
image = /vmlinuz
  root = /dev/hda1
  label = linux
# Linux bootable partition config ends
```

5. Test lilo and watch for error messages:
   lilo -tv
6. Run lilo:
   lilo -v
7. exit the chroot shell

   Now your / directory again is in the memory filesystem.
   Directory /mnt is the root directory of the harddrive filesystem.

11.11 Confirm Disk Contents

1. ls -al /mnt
   Check for bin, boot, dev, etc, lib, root, sbin, usr, var directories.
   These are expanded in step 11.6.1 from scd.linux.tgz.
2. ls -al /mnt
   Check for proc, cdrom, mnt, and tmp direcories.
   These are expanded in step 11.6.1 from scd.linux.tgz.
   If necessary these can be created with mkdir command.
3. Check for 't' attribute on tmp directory.
   This can be added with chmod +t /mnt/tmp
4. ls -al /mnt/home
   Check for scdadmin, scdops, scdcm, scdtest, scdinstl directories
   These are expanded in step 6.2 from scd.home.tgz.
   Rerun step 6.2 if necessary.
   Check that file ownership is not root but the userid.
5. more /mnt/etc/fstab
   and check that swap and root systems are set properly.
6. more /mnt/mnt/etc/fstab (assuming that hda3 is still mounted !)
   and check that the swap and root systems are set properly on the alternate partition.
7. repeat checks 1, 2, 3, 4 on the other partition mounted under /mnt.

11.12 Reboot the SCD from Hard Drive

1. Remove any floppy from the diskette drive and
   Type ALT+CNTL+DELETE to reboot the host.
2. The system should boot on the primary partition hda2.
3. If reboot fails:
   Boot with floppy disks (step 1).
   Mount the hard drive (step 4).
   Validate and run lilo procedure (step 9).
4. login as root
   Check that password matches expected delivery of passwords.
   There may be a newer version of scd.etc.tgz.
5. login as other users: scdadmin, scdcm, scdops, scdtest
   and validate password and startup script.
6. Check /home/scdcm/scd/scripts/scntr.init for the correct fileserver addresses.
7. Run ifconfig and verify Ethernet configuration

```
lo   Link encap:Local Loopback
     inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
     UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
     RX packets:66839 errors:0 dropped:0 overruns:0
     TX packets:66839 errors:0 dropped:0 overruns:0

eth0 Link encap:10Mbps Ethernet  HWaddr 00:A0:24:80:2C:D6
     inet addr:192.168.201.116  Bcast:192.168.201.255 Mask:255.255.255.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:283772 errors:0 dropped:0 overruns:0
```

TX packets:2489 errors:0 dropped:0 overruns:0
Interrupt:11 Base address:0xef80

Check file /etc/rc.d/rc.local for correct startup procedure.
If missing eth0 entirely, check CMOS setup. IRQ3,4,5 should be set for ISA/EISA.
Validate Ethernet card with DOS floppy and 3COM test program.

11.13 Reset the BIOS

1. Insert Boot floppy and depress hardware reset button, or power on.
2. Depress the DEL key during bootup to go into BIOS CMOS Setup
3. Tab to the 'advanced' menu and change the 'BootUp Sequence' to 'C:,A::CDROM'
   so the system will boot from the floppy drive.
   Remember that this BIOS setting must be set back to 'C:A:CDROM' when the SCD is in use. If not, the SCD coldstart procedures will hang when there is a SCD Configuration floppy that in the disk drive.
4. Use ESCape to leave BIOS setup and save the settings.

Appendix A - Parts List

A.1 Lockheed/Martin Built Chassis - Redundant Power Supply 85-62805-xx

NSN	Manu. Num	Part Num	Description
5975-00-N96-3490	OUY60	APRE-4001	ENCLOSURE, RACK MOUNT, REDUNDANT PS
6130-00-N96-3536	OUY60	PS-4000	POWER SUPPLY, REDUNDANT
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.2 Lockheed/Martin Built Chassis - Non-Redundant Power Supply 85-62806-xx

NSN	Manu. Num	Part Num	Description
5975-00-N96-3491	OUY60	APRE-4008	ENCLOSURE, RACK MOUNT, NON-REDU PS
6130-00-N96-3492	OUY60	PS-5002	POWER SUPPLY, NON-REDUNDANT
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.3 APPRO Built Chassis - Redundant Power Supply 85-62794-xx

NSN	Manu. Num	Part Num	Description
7025-00-N96-3740	OUY60	SYS-R5M01-VCAT12MBV-4001PS4000-4XCD	ENCLOSURE, RACK MOUNT, REDUNDANT PS (INCLUDES ALL OF THE PARTS LISTED BELOW)
6130-00-N96-3536	OUY60	PS-4000	POWER SUPPLY, REDUNDANT
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.4 APPRO Built Chassis - Non-Redundant Power Supply 85-62793-xx

NSN	Manu. Num	Part Num	Description
7025-00-N96-3741	OUY60	SYS-R5M01-VCAT12MBV-4008PS5002-4XCD	ENCLOSURE, RACK MOUNT, NON-REDU PS (INCLUDES ALL OF THE PARTS LISTED BELOW)
6130-00-N96-3492	OUY60	PS-5002	POWER SUPPLY, NON-REDUNDANT
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.5 AVTEC Built Chassis - Redundant Power Supply 85-62917-xx

NSN	Manu. Num	Part Num	Description
7025-00-N96-3947	6X005	PTP-1001	ENCLOSURE, RACK MOUNT, REDUNDANT PS
7025-00-N96-3946	6X005	PTP-9000	POWER SUPPLY, REDUNDANT
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.6 AVTEC Built Chassis - Non-Redundant Power Supply 85-62918-xx

NSN	Manu. Num	Part Num	Description
7025-00-N96-3948	6X005	PTP-1001-NON-RED	ENCLOSURE, RACK MOUNT, NON-REDU PS
7025-00-N96-3946	6X005	PTP-9000	POWER SUPPLY, NON-REDUNDANT
6130-00-R96-3019	6X005	PTP-9001	PS OPTION, NON-REDUNDANT, -48VDC
7025-00-N95-3292	DO113	S1468	MOTHERBOARD, PC COMP. W/32MB MEM
7025-00-N95-7425	52840	WD2850A	HARD DRIVE, IDE, 850MB
7025-00-N96-3232	50356	FD-235HF	FLOPPY DRIVE, 3.5''
7025-00-N96-3744	OUY60	CD-MIEI01	OPTICAL DRIVE

A.7 PC Cards

NSN	Manu. Num	Part Num	Description
7025-00-N95-7862	53450	109-25500-40	VIDEO CONTROLLER, PRO TURBO
7025-00-R95-3315	53450	XPRESSION	VIDEO CONTROLLER, EXPRESSION
7025-00-N96-2776	IDF36	3C900 COMBO	INTERFACE CARD, ETHERNET
7025-00-N92-1470	25306	1486797	NIB CARD - SERIAL I/F FOR SCD
7025-00-N93-0516	6X005	AT3010	AVTEC CARD - SERIAL I/F FOR PTP
TBS	TBS	P56-1023	PNY-MEMORY, 16MB, 60NS, EDO, 72-PIN SIMM

A.8 Peripherals

NSN	Manu. Num	Part Num	Description
7025-00-N96-3743	OUY60	KB-CHM01	KEYBOARD (RACK MOUNT REPLACEMENT)
7025-00-R95-3163	0EXA9	471320	MICROSOFT MOUSE, SERIAL W/9-PIN INTERFACE
7025-00-N93-3931	OUY60	1451-C	CTX 14’’ COLOR MONITOR
7025-00-N96-3487	OUY60	RSB-03001	SWITCH FOR MONITOR
7025-00-R96-3037	OBC68	TPT-D4	TRANSCEIVER, REDUNDANT